

Bronze_customer.sql

```
CREATE OR REFRESH STREAMING TABLE
${catalog_name}.${bronze_schema}.bronze_customer
COMMENT 'Bronze streaming table with raw data and inferred schema'
TBLPROPERTIES ('quality' = 'bronze')
AS SELECT *,
  _metadata AS source_metadata
FROM STREAM
read_files('${customer_ingest_volume}');
```

Bronze_location.sql

```
CREATE OR REFRESH STREAMING TABLE
${catalog_name}.${bronze_schema}.bronze_location
COMMENT 'Bronze streaming table with raw data and inferred schema'
TBLPROPERTIES ('quality' = 'bronze')
AS SELECT *,
  _metadata AS source_metadata
FROM STREAM
read_files('${location_ingest_volume}');
```

Bronze_product.sql

```
CREATE OR REFRESH STREAMING TABLE
${catalog_name}.${bronze_schema}.bronze_product
COMMENT 'Bronze streaming table with raw data and inferred schema'
TBLPROPERTIES ('quality' = 'bronze')
AS SELECT *,
  _metadata AS source_metadata
FROM STREAM
read_files('${product_ingest_volume}');
```

Bronze_sales.sql

```
CREATE OR REFRESH STREAMING TABLE ${catalog_name}.${bronze_schema}.bronze_sales
COMMENT 'Bronze streaming table with raw data and inferred schema'
TBLPROPERTIES ('quality' = 'bronze')
AS SELECT *,
  _metadata AS source_metadata
```

```
FROM STREAM
read_files('${sales_ingest_volume}');
```

Silver_customer.sql

```
CREATE OR REFRESH STREAMING TABLE
${catalog_name}.${silver_schema}.silver_customer (
  customer_id          INT NOT NULL,
  customer_type        STRING NOT NULL,
  name                 STRING NOT NULL,
  first_purchase_date  DATE NOT NULL,
  location_id          INT NOT NULL,
  CONSTRAINT customer_pk PRIMARY KEY (customer_id)
  --FOREIGN KEY (location_id) REFERENCES
${catalog}.${silver_schema}.silver_location(location_id)
)
COMMENT 'Silver streaming table with renamed columns and enforced datatypes'
TBLPROPERTIES ('quality' = 'silver')
AS
SELECT
  CAST(customerid AS INT) AS customer_id,
  CAST(customer_type AS STRING),
  CAST(name AS STRING),
  CAST(first_purchase_date AS DATE),
  CAST(locationid AS INT) AS location_id
FROM STREAM(
  ${catalog_name}.${bronze_schema}.bronze_customer
)
WHERE
  customerid IS NOT NULL AND
  customer_type IS NOT NULL AND
  customer_type != 'null' AND
  name IS NOT NULL AND
  name != 'null' AND
  first_purchase_date IS NOT NULL AND
  locationid IS NOT NULL;
```

Silver_location.sql

```

CREATE OR REFRESH STREAMING TABLE
${catalog_name}.${silver_schema}.silver_location (
    location_id          INT NOT NULL,
    city                 STRING,
    state                STRING,
    country              STRING,
    CONSTRAINT location_id PRIMARY KEY (location_id)
)
COMMENT 'Silver streaming table with renamed columns and enforced datatypes'
TBLPROPERTIES ('quality' = 'silver')
AS
SELECT
    CAST(locationid AS INT) AS location_id,
    CAST(city AS STRING),
    CAST(state AS STRING),
    CAST(country AS STRING)
FROM STREAM(
    ${catalog_name}.${bronze_schema}.bronze_location
)
WHERE
    locationid IS NOT NULL AND
    city != 'null' AND
    state != 'null' AND
    country != 'null';

```

Silver_product.sql

```

CREATE OR REFRESH STREAMING TABLE
${catalog_name}.${silver_schema}.silver_product (
    product_id          INT NOT NULL,
    name                 STRING NOT NULL,
    base_price           DOUBLE NOT NULL,
    brand                STRING NOT NULL,
    category             STRING NOT NULL,
    description          STRING NOT NULL,
    subbrand             STRING NOT NULL,
    CONSTRAINT product_id PRIMARY KEY (product_id)
)
COMMENT 'Silver streaming table with renamed columns and enforced datatypes'
TBLPROPERTIES ('quality' = 'silver')
AS
SELECT
    CAST(productid AS INT) AS product_id,
    CAST(name AS STRING),
    CAST(base_price AS DOUBLE),
    CAST(brand AS STRING),
    CAST(category AS STRING),
    CAST(description AS STRING),
    CAST(subbrand AS STRING)
FROM STREAM(
    ${catalog_name}.${bronze_schema}.bronze_product
)
WHERE
    productid IS NOT NULL AND
    name != 'null' AND
    base_price IS NOT NULL AND
    brand != 'null' AND
    category != 'null' AND
    description != 'null' AND
    subbrand != 'null';

```

Silver_sales.sql

```

CREATE OR REFRESH STREAMING TABLE ${catalog_name}.${silver_schema}.silver_sales
(
    customer_id          INT NOT NULL,
    date_of_sale         DATE NOT NULL,
    order_status         STRING NOT NULL,
    payment_method       STRING NOT NULL,
    product_id           INT NOT NULL,
    quantity             INT NOT NULL,
    sales_id             INT NOT NULL,
    total_sales          DOUBLE NOT NULL,
    unit_price           DOUBLE NOT NULL,
    CONSTRAINT sales_pk PRIMARY KEY (sales_id)
    --FOREIGN KEY (customer_id) REFERENCES
${catalog}.${silver_schema}.silver_customer(customer_id),
    --FOREIGN KEY (product_id) REFERENCES
${catalog}.${silver_schema}.silver_product(product_id)
)
COMMENT 'Silver streaming table with renamed columns and enforced datatypes'
TBLPROPERTIES ('quality' = 'silver')
AS
SELECT
    CAST(customerid AS INT) AS customer_id,
    CAST(date_of_sale AS DATE),
    CAST(order_status AS STRING),
    CAST(payment_method AS STRING),
    CAST(productid AS INT) AS product_id,
    CAST(quantity AS INT),
    CAST(salesid AS INT) AS sales_id,
    CAST(total_sales AS DOUBLE),
    CAST(unit_price AS DOUBLE)
FROM STREAM(
    ${catalog_name}.${bronze_schema}.bronze_sales
)
WHERE
    customerid IS NOT NULL AND
    date_of_sale IS NOT NULL AND
    order_status != 'null' AND
    payment_method != 'null' AND

```

```
productid IS NOT NULL AND
quantity IS NOT NULL AND
salesid IS NOT NULL AND
total_sales IS NOT NULL AND
unit_price IS NOT NULL;
```

Gold_spend_per_country.sql

```
CREATE MATERIALIZED VIEW ${catalog_name}.${gold_schema}.gold_spend_per_country
COMMENT 'Gold table with sales data to determine total spend by each country'
TBLPROPERTIES ('quality' = 'gold')
AS
with cte AS (
  SELECT
    sales.date_of_sale,
    sales.total_sales,
    sales.customer_id,
    customers.location_id
  FROM ${catalog_name}.${silver_schema}.silver_sales AS sales
  LEFT JOIN (SELECT customer_id, location_id FROM
${catalog_name}.${silver_schema}.silver_customer) AS customers
    ON sales.customer_id = customers.customer_id
)

SELECT
  YEAR(date_of_sale) AS year,
  country,
  ROUND(SUM(total_sales),2) AS total_sales
FROM cte
LEFT JOIN (SELECT location_id, country FROM
${catalog_name}.${silver_schema}.silver_location) as loc
ON cte.location_id = loc.location_id
GROUP BY year, country
```

Gold_top_product_by_month.sql

```

CREATE MATERIALIZED VIEW
${catalog_name}.${gold_schema}.gold_top_product_each_month
COMMENT 'Gold table with sales data to determine the top selling product each
month'
TBLPROPERTIES ('quality' = 'gold')
AS
SELECT
    sales.product_id,
    MONTH(sales.date_of_sale) AS month,
    YEAR(sales.date_of_sale) AS year,
    products.name as product_name,
    SUM(sales.quantity) AS quantity,
    ROUND(SUM(sales.quantity * products.base_price), 2) AS revenue
FROM ${catalog_name}.${silver_schema}.silver_sales sales
LEFT JOIN ${catalog_name}.${silver_schema}.silver_product products
ON sales.product_id == products.product_id
GROUP BY sales.product_id, products.name, month, year
ORDER BY revenue DESC

```

ML_data.sql

```

CREATE OR REPLACE MATERIALIZED VIEW ${catalog_name}.${silver_schema}.ml_data (
    composite_key    STRING NOT NULL PRIMARY KEY,
    month            INT,
    year            INT,
    total_sales      DOUBLE,
    product_name     STRING
)
COMMENT 'ML table with sales data for ML enablement'
TBLPROPERTIES ('quality' = 'gold')
AS
SELECT
    CONCAT(YEAR(date_of_sale), '-', MONTH(date_of_sale)) AS composite_key,
    MONTH(date_of_sale) AS month,
    YEAR(date_of_sale) AS year,
    ROUND(SUM(total_sales), 2) AS total_sales,
    prod.name AS product_name
FROM ${catalog_name}.${silver_schema}.silver_sales sale

```

```
LEFT JOIN ${catalog_name}.${silver_schema}.silver_product prod
WHERE sale.product_id = prod.product_id
GROUP BY product_name, MONTH(date_of_sale), YEAR(date_of_sale)
```